CLAIMS

What is claimed is:

1. A system to provide a software debugging environment, comprising:

   an executing software program containing at least one data structure;

   at least one abstract view capable of displaying and/or editing at least one abstract content of the at least one data structure; and

   at least one filter capable of defining a displaying and/or editing property of the at least one abstract view, such property can include at least one of: which of the at least one abstract content is displayed, a format in which it is displayed, and how it is edited.

2. The system according to claim 1, wherein:

   the system is at least partially implemented using Java language.

3. The system according to claim 1, further comprising:

   at least one editor associated with the at least one abstract view capable of at least one of:

   allowing the at least one abstract content to be modified through the at least one abstract view; and

   validating an input value to the at least one abstract content against an allowed value for the at least one abstract content.

4. The system according to claim 1, wherein:

   the at least one abstract view is capable of presenting the at least one abstract content of the at least one data structure without showing a physical implementation of the at least one data structure.

5. The system according to claim 1, wherein:

   each of the at least one abstract view can be individually selected for display.

6. The system according to claim 1, wherein:

   Two or more of the at least one abstract view are capable of displaying and/or editing the same one of the at least one abstract content without being deadlocked.

7. The system according to claim 1, wherein:

the at least one filter can be defined via configuration information stored in a file, which can be an XML file.

8. The system according to claim 1, further comprising:

a component capable of interactively performing at least one of:

selecting a subset of the at least one of abstract view for display; and

defining the displaying and/or editing property of the at least one filter.

9. The system according to claim 8, wherein:

the component can be realized via an interface to an Integrated Development Environment (IDE).

10. The system according to claim 1, further comprising:

at least one component capable of supporting the debugging of a JSP page and a machine generated servlet that implements the JSP page.

11. The system according to claim 10, wherein:

the at least one component can perform at least one of:

extracting and displaying a code and/or a content of interest, and mapping them to a format used in a source code in a JSP page, for use with executing a JSP servlet;

following an execution path through at least one level of redirection using at least one tag;

extracting and manipulating a streaming data from a content of a buffer used to transmit and receive the streaming data; and

setting at least one break point in a JSP page and stepping through the execution of the page based on the displaying property.

12. The system according to claim 11, wherein:

the streaming data can be extracted by inserting a wrapper or "writer" class around the JSP servlet.

13. A method to provide a software debugging environment, comprising:

displaying and/or editing at least one abstract content of at least one data structure in an executing software program via at least one abstract view; and

defining a displaying and/or editing property of the at least one abstract view via at least one filter, such property can include at least one of: which of the at least one abstract content is displayed, a format in which it is displayed, and how it is edited.

14. The method according to claim 13, further comprising:

allowing the at least one abstract content to be modified through the at least one abstract view; and

validating an input value to the at least one content against an allowed value for the at least one content.

15. The method according to claim 13, further comprising:

presenting the at least one abstract content of the at least one data structure without showing a physical implementation of the at least one data structure.

16. The method according to claim 13, further comprising:

selecting each of the at least one abstract view individually for display.

17. The method according to claim 13, further comprising:

displaying and/or editing the same one of the at least one abstract content via two or more of the at least one abstract view without being deadlocked.

18. The method according to claim 13, further comprising:

defining the at least one filter via configuration information stored in a file, which can be an XML file.

19. The method according to claim 13, further comprising:

interactively performing at least one of:

selecting a subset of the at least one of abstract view for display; and

defining the displaying and/or editing property of the at least one filter.

20. The method according to claim 13, further comprising:

supporting the debugging of a JSP page and a machine generated servlet that implements the JSP page.

21. The method according to claim 20, further comprising:

extracting and displaying a code and/or a content of interest, and mapping them to a format used in a source code in a JSP page, for use with executing a JSP servlet;

following an execution path through at least one level of redirection using at least one tag;

extracting and manipulating a streaming data from a content of a buffer used to transmit and receive the streaming data; and

setting at least one break point in a JSP page and stepping through the execution of the page based on the displaying property.

22. The method according to claim 21, wherein:

the streaming data can be extracted by inserting a wrapper or "writer" class around the JSP servlet.

23. A machine readable medium having instructions stored thereon that when executed by a processor cause a system to:

display and/or edit at least one abstract content of at least one data structure in an executing software program via at least one abstract view; and

define a displaying and/or editing property of the at least one abstract view via at least one filter, such property can include at least one of: which of the at least one abstract content is displayed, a format in which it is displayed, and how it is edited.

24. The machine readable medium of claim 23, further comprising instructions that when executed cause the system to:

allow the at least one abstract content to be modified through the at least one abstract view; and

validate an input value to the at least one abstract content against an allowed value for the at least one content.

25. The machine readable medium of claim 23, further comprising instructions that when executed cause the system to:

present the at least one abstract content of the at least one data structure without showing an physical implementation of the at least one data structure.

26. The machine readable medium of claim 23, further comprising instructions that when executed cause the system to:

select each of the at least one abstract view individually for display.

27. The machine readable medium of claim 23, further comprising instructions that when executed cause the system to:

display and/or edit the same one of the at least one abstract content via two or more of the at least one abstract view without being deadlocked.

28. The machine readable medium of claim 23, further comprising instructions that when executed cause the system to:

define the at least one filter via configuration information stored in a file, which can be an XML file.

29. The machine readable medium of claim 23, further comprising instructions that when executed cause the system to:

interactively perform at least one of:

selecting a subset of the at least one of abstract view for display; and

defining the displaying and/or editing property of the at least one filter.

30. The machine readable medium of claim 23 further comprising instructions that when executed cause the system to:

support the debugging of a JSP page and a machine generated servlet that implements the JSP page.

31. The machine readable medium of claim 30, further comprising instructions that when executed cause the system to:

extract and display a code and/or a content of interest, and mapping them to a format used in a source code in a JSP page, for use with executing a JSP servlet;

follow an execution path through at least one level of redirection using at least one tag;

extract and manipulate a streaming data from a content of a buffer used to transmit and receive the streaming data; and

set at least one break point in a JSP page and step through the execution of the page based on the displaying property.

32. The machine readable medium of claim 31, wherein:

the streaming data can be extracted by inserting a wrapper or "writer" class around the JSP servlet.

33. A system to provide a software debugging environment, comprising:

means for displaying and/or editing at least one abstract content of at least one data structure in an executing software program via at least one abstract view; and

means for defining a displaying and/or editing property of the at least one abstract view via at least one filter, such property can include at least one of: which of the at least one abstract content is displayed, a format in which it is displayed, and how it is edited.

34. A computer data signal embodied in a transmission medium, comprising:

a code segment including instructions to display and/or edit at least one abstract content of at least one data structure in an executing software program via at least one abstract view; and

a code segment including instructions to define a displaying and/or editing property of the at least one abstract view via at least one filter, such property can include at least one of: which of the at least one abstract content is displayed, a format in which it is displayed, and how it is edited.